

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号
特開2003-296123
(P2003-296123A)

(43) 公開日 平成15年10月17日 (2003. 10. 17)

(51) Int.Cl. ⁷	識別記号	F I	テーマコード* (参考)
G 0 6 F 9/45		G 0 6 F 9/30	3 5 0 A 5 B 0 1 1
1/32		15/78	5 1 0 P 5 B 0 3 3
9/30	3 5 0	9/44	3 2 2 F 5 B 0 6 2
9/318		1/00	3 3 2 B 5 B 0 8 1
15/78	5 1 0	9/30	3 2 0 C
審査請求 未請求 請求項の数25 O L (全 12 頁)			

(21) 出願番号 特願2003-17374(P2003-17374)
(22) 出願日 平成15年 1 月27日 (2003. 1. 27)
(31) 優先権主張番号 特願2002-21479(P2002-21479)
(32) 優先日 平成14年 1 月30日 (2002. 1. 30)
(33) 優先権主張国 日本 (J P)

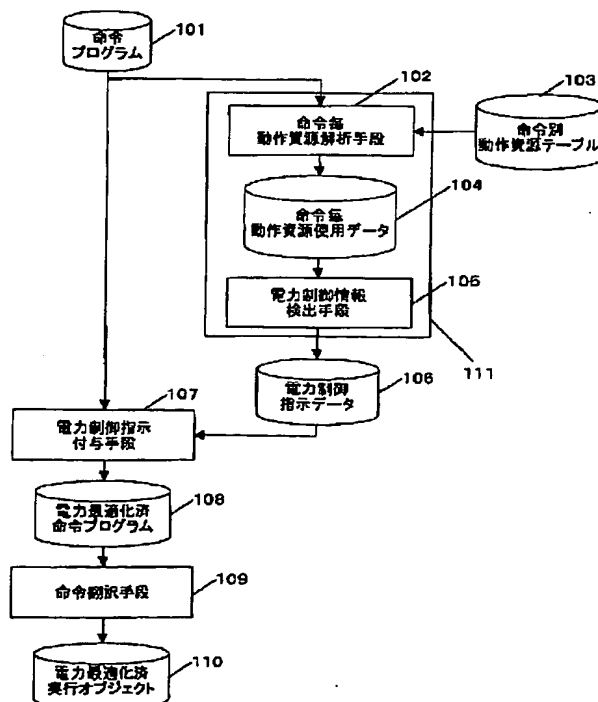
(71) 出願人 000005821
松下電器産業株式会社
大阪府門真市大字門真1006番地
(72) 発明者 谷 丈暢
大阪府門真市大字門真1006番地 松下電器
産業株式会社内
(74) 代理人 100097445
弁理士 岩橋 文雄 (外 2 名)
Fターム(参考) 5B011 EA08 LL06 LL11
5B033 BE00
5B062 AA05 DD10 HH06 HH07
5B081 AA07 CC00

(54) 【発明の名称】 電力制御情報を付与する命令変換装置及び命令変換方法、命令変換を実現するプログラム及び回路、変換された命令を実行するマイクロプロセッサ

(57) 【要約】

【課題】 マイクロプロセッサにおいて、ハードウェアの規模を増大することなく、またユーザーの開発負担を伴わず、ソフトウェアの汎用性を確保しながら、きめ細かな電力制御を行うことを課題とする。

【解決手段】 命令の種類毎に、マイクロプロセッサ動作時に各動作資源が動作するか否かに関する情報を予めテーブルに保持する。このテーブルを参照し、マイクロプロセッサのプログラム実行時に所定命令区間動作しない動作資源を検出し、電力制御指示データを出力する。この電力制御指示データを参照し、命令プログラムを記述してから実行オブジェクトがデコードするまでの間に、命令プログラムあるいは実行オブジェクトに電力制御に関する情報を直接付与する。



【特許請求の範囲】

【請求項1】 複数の命令から構成される命令プログラムを所定のマイクロプロセッサの実行に適したように最適化する命令変換装置であって、

前記マイクロプロセッサ動作時に所定の長さの命令区間動作しない動作資源を検出する電力制御情報解析手段と、

前記電力制御情報解析手段の検出結果に基づいて、前記命令プログラムに、電力制御に関する指示を付与する電力制御指示付与手段とを備えることを特徴とする命令変換装置。

【請求項2】 請求項1記載の命令変換装置であって、前記命令プログラムを構成する命令の種類により、各動作資源が動作するか否かに関する情報が格納されている命令別動作資源テーブル格納手段とをさらに備え、前記電力制御情報解析手段は、前記命令別動作資源テーブル格納手段に格納された情報に基づいて、前記マイクロプロセッサ動作時に所定の長さの命令区間動作しない動作資源を検出することを特徴とする命令変換装置。

【請求項3】 請求項2記載の命令変換装置であって、前記電力制御指示付与手段は、前記命令プログラムに、動作資源の停止あるいは動作開始の命令を挿入することにより、電力制御に関する情報を付与することを特徴とする命令変換装置。

【請求項4】 請求項2記載の命令変換装置であって、前記電力制御指示付与手段は、前記命令プログラムにおいて、動作資源が停止可能な命令区間の直前に動作資源の停止の命令を挿入し、動作資源が停止可能な命令区間の最後に動作資源の動作開始の命令を挿入することにより、電力制御に関する指示を付与することを特徴とする命令変換装置。

【請求項5】 請求項2記載の命令変換装置であって、前記電力制御指示付与手段は、前記命令プログラム中の一の命令を、動作資源の停止あるいは動作開始の指示を含む一の命令に置換することにより、電力制御に関する指示を付与することを特徴とする命令変換装置。

【請求項6】 請求項2記載の命令変換装置であって、前記電力制御指示付与手段は、前記命令プログラムにおいて、動作資源が停止可能な命令区間の直前の命令を動作資源の停止の指示を含む命令に置換し、動作資源が停止可能な命令区間の最後の命令を動作資源の動作開始の指示を含む命令に置換することにより、電力制御に関する指示を付与することを特徴とする命令変換装置。

【請求項7】 請求項1記載の命令変換装置であって、予め設定された電力制御のレベルを判定し、前記電力制御情報解析手段へと判定結果を出力する電力制御レベル判定手段とをさらに備え、前記電力制御情報解析手段は、前記電力制御レベル判定手段の判定結果を加味して、電力制御可能な命令区間を検出することを特徴とする命令変換装置。

【請求項8】 請求項7記載の命令変換装置であって、前記電力制御のレベルは、前記所定の長さに対応しており、

前記電力制御情報解析手段は、前記電力制御レベル判定手段の判定結果に応じて、前記所定の長さを切り換えることを特徴とする命令変換装置。

【請求項9】 請求項7記載の命令変換装置であって、前記電力制御のレベルは、電力制御の対象となる動作資源の種類に対応しており、

前記電力制御情報解析手段は、前記電力制御レベル判定手段の判定結果に応じて、検出の対象となる前記動作資源の種類を切り換えることを特徴とする命令変換装置。

【請求項10】 請求項1記載の命令変換装置であって、

予め設定された電力制御のレベルを判定し、前記電力制御情報解析手段へと判定結果を出力する電力制御レベル判定手段とをさらに備え、

前記電力制御情報解析手段は、前記電力制御レベル判定手段の判定結果に応じて、前記所定の長さを切り換えることを特徴とする命令変換装置。

【請求項11】 請求項1記載の命令変換装置で変換された命令プログラムを実行することを特徴とするマイクロプロセッサ。

【請求項12】 複数の命令から構成される命令プログラムを所定のマイクロプロセッサの実行に適したように最適化する命令変換方法であって、

前記マイクロプロセッサ動作時に所定の長さの命令区間動作しない動作資源を検出する電力制御情報解析ステップと、

前記電力制御情報解析ステップにおける検出結果に基づいて、前記命令プログラムに、電力制御に関する指示を付与する電力制御指示付与ステップとを備えることを特徴とする命令変換方法。

【請求項13】 複数の命令から構成される命令プログラムを所定のマイクロプロセッサの実行に適したように最適化する命令変換方法であって、

予め設定された電力制御のレベルを判定する電力制御レベル判定ステップと、

前記電力制御レベル判定ステップにおける判定結果を加味して、前記マイクロプロセッサ動作時に所定の長さの命令区間動作しない動作資源を検出する電力制御情報解析ステップと、

前記電力制御情報解析ステップにおける検出結果に基づいて、前記命令プログラムに、電力制御に関する指示を付与する電力制御指示付与ステップとを備えることを特徴とする命令変換方法。

【請求項14】 請求項13記載の命令変換方法であって、前記電力制御のレベルは、前記所定の長さに対応しており、

前記電力制御情報解析ステップでは、前記電力制御レベル判定ステップにおける判定結果に応じて、前記所定の長さを切り換えることを特徴とする命令変換方法。

【請求項 15】 請求項 13 記載の命令変換方法であって、

前記電力制御のレベルは、電力制御の対象となる動作資源の種類に対応しており、

前記電力制御情報解析ステップでは、前記電力制御レベル判定ステップにおける判定結果に応じて、検出の対象となる前記動作資源の種類を切り換えることを特徴とする命令変換方法。

【請求項 16】 複数の命令から構成される命令プログラムを所定のマイクロプロセッサの実行に適したように最適化する命令変換方法であって、

前記命令プログラムを構成する命令の種類により、各動作資源が動作するか否かに関する情報を予め命令別動作資源テーブルに記録することを特徴とする命令変換方法。

【請求項 17】 請求項 16 記載の命令変換方法であって、

前記命令別動作資源テーブルを参照し、マイクロプロセッサ動作時に所定命令区間動作しない動作資源を検出するステップと、

前記検出結果に基づいて、前記命令プログラムに、電力制御に関する指示を付与するステップとを備えることを特徴とする命令変換方法。

【請求項 18】 請求項 17 記載の命令変換方法であって、

前記命令プログラムに、動作資源の停止あるいは動作開始の命令を挿入することにより、電力制御に関する指示を付与することを特徴とする命令変換方法。

【請求項 19】 請求項 17 記載の命令変換方法であって、

前記命令プログラムにおいて、動作資源が停止可能な命令区間の直前に動作資源の停止の命令を挿入し、動作資源が停止可能な命令区間の最後に動作資源の動作開始の命令を挿入することにより、電力制御に関する指示を付与することを特徴とする命令変換方法。

【請求項 20】 請求項 17 記載の命令変換方法であって、

前記命令プログラム中の一の命令を、動作資源の停止あるいは動作開始の指示を含む一の命令に置換することにより、電力制御に関する指示を付与することを特徴とする命令変換方法。

【請求項 21】 請求項 17 記載の命令変換方法であって、

前記命令プログラムにおいて、動作資源が停止可能な命令区間の直前の命令を動作資源の停止の指示を含む命令に置換し、動作資源が停止可能な命令区間の最後の命令を動作資源の動作開始の指示を含む命令に置換すること

により、電力制御に関する指示を付与することを特徴とする命令変換方法。

【請求項 22】 マイクロプロセッサが実行する実行オブジェクトを構成する命令コードの種類により、各動作資源が動作するか否かに関する情報を予め記録した命令別動作資源メモリと、

前記命令別動作資源メモリの情報に基づき、マイクロプロセッサ動作時に所定命令区間動作しない動作資源を検出する電力制御情報解析回路と、

前記命令別動作資源解析回路の検出結果を基に、実行オブジェクトに、電力制御に関する指示を付与する電力制御指示付与回路とを備える命令変換回路。

【請求項 23】 請求項 22 記載の命令変換回路を備えることを特徴とするマイクロプロセッサ。

【請求項 24】 請求項 11 記載のマイクロプロセッサであって、

前記変換された命令プログラムをデコードする命令デコーダと、

前記命令デコーダのデコード結果に基づいて、電力制御に関する情報が書き込まれる電力制御レジスタと、

割り込み・分岐・例外の少なくとも 1 つを検出し、検出されたときに、前記電力制御レジスタの状態を、全ての動作資源が動作する状態に設定する状態リセット回路とを備えることを特徴とするマイクロプロセッサ。

【請求項 25】 コンピュータを、

マイクロプロセッサが実行する命令プログラムを構成する命令の種類により、各動作資源が動作するか否かに関する情報を予め記録した命令別動作資源テーブルを参照し、マイクロプロセッサ動作時に所定命令区間動作しない動作資源を検出する電力制御情報解析手段、前記命令別動作資源解析手段の検出結果を基に、命令プログラムに、電力制御に関する指示を付与する電力制御指示付与手段として機能させるためのプログラム。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は、実行すべき実行オブジェクトを命令記憶装置（命令メモリ・命令キャッシュ等）から取得し、これをデコーダで解読し、その内容により動作資源（演算装置や入力装置や出力装置や記憶装置や制御装置）を制御し順序処理を進めるマイクロプロセッサ（マイクロコンピュータやマイクロコントローラやデジタル・シグナル・プロセッサを含む）において、実行すべき実行オブジェクトに電力制御情報を付与する命令変換装置及び命令変換方法、命令変換を実現するプログラム及び回路、さらには変換された命令を実行するマイクロプロセッサに関するものである。

【0002】

【従来の技術】従来、マイクロプロセッサにおいて、低消費電力を目的に電力の制御を行う方法は二つあった。一つはハードウェア駆動による電力制御方法であり、一

つはソフトウェア駆動による電力制御方法である。

【0003】まず、従来のハードウェア駆動による電力制御方法を説明する。通常、マイクロプロセッサの動作では、プログラム進行に応じて、命令デコーダが命令記憶装置から命令コードを取得し、その命令コードをデコードすることにより、各命令に応じた演算等の動作制御を行う。ハードウェア駆動により消費電力を制御する場合、電力制御に関する指示を抽出するための命令デコーダをマイクロプロセッサに組み込むことにより、電力制御を行う。

【0004】図1.4はハードウェア駆動により、マイクロプロセッサの電力制御を行う回路方式の例である。命令デコーダ1401は、レジスタ間の加算命令の命令コードを命令記憶装置1402から取得すると、該当命令コード実行中はデータメモリや周辺インターフェース等を使用しないと判断する。命令デコーダ1401はレジスタ間の加算命令を取得する毎に、データメモリや周辺インターフェース等を停止するように、各回路に指示を出す。これにより、レジスタ間の加算命令に対しては電力制御を行うことができる。他に減算命令や読み出し命令など、複数の命令コードの種類に応じて電力制御に関する指示をデコード可能なようにデコーダを構成する。これによって、さまざまな命令コードをデコードする毎に電力制御の指示を行い、消費電力の削減を行う。

【0005】次に、従来のソフトウェア駆動による電力制御方法を説明する。ソフトウェア駆動ではプログラムを記述するユーザーが、マイクロプロセッサが命令プログラムを実行している際に停止可能な動作資源を判断して、プログラム中に電力制御に関する命令を直接記述する。

【0006】図1.5はソフトウェア駆動により、マイクロプロセッサの電力制御を行うプログラムの例を表わしたものである。本例では区間1501においてデータメモリを使用していない。ユーザーは1502の個所にデータメモリを停止させる命令を挿入し、データメモリが再び使用される直前である1503にデータメモリの停止を解除する命令を挿入する。前記2つの命令により、区間1501においてデータメモリを停止させることが可能である。本例ではデータメモリの例を示したが、データレジスタ、データバス、周辺インターフェース、さらにはAND等の回路素子レベルに及ぶ、マイクロプロセッサのあらゆる回路が制御可能である。ソフトウェア駆動において電力制御を行う場合であっても、命令デコーダでデコードされた電力制御に関する指示を電力制御レジスタに書き込むことにより、マイクロプロセッサ内の電力制御を行う。

【0007】なお、本発明に係る先行技術として電力制御情報を書き換え可能に記憶する電力制御レジスタを備えたパワーテーブルを設け、複数の動作条件を電力制御レジスタに記憶させ、プロセッサの現在の動作が複

数の動作条件のいずれを満たすかを判定し、判定の結果に応じて電力制御を行うものがある（例えば特許文献1参照）。

【0008】

【特許文献1】特開2002-182807号公報（第2図、第5図）

【0009】

【発明が解決しようとする課題】ハードウェア駆動による電力制御方法では、各命令コードに応じた電力制御情報をデコード可能なようにデコーダを構成する必要があるため、きめ細かな電力制御を行おうとするとハードウェア規模が大きくなる。また、デコード可能な命令の種類が多いと、デコード動作による消費電力が増加する。そのため、かえってプロセッサとしての消費電力も増加してしまう恐れがある。さらに、電力制御情報は、命令コード単位でしか抽出できないため、後続の命令に対する先行した電力制御は行えない。

【0010】ソフトウェア駆動による電力制御方法では、ユーザーがプログラム記述時に電力制御の開始や終了個所を判断し、直接プログラムに電力制御用命令の記述を行うため、ユーザーの開発負担が大きい。また、指定個所が増加する場合、きめ細かな制御指示を記述するのは困難である。また、電力制御の対象がメモリ構成や周辺インターフェースポート数等、マイクロプロセッサの構成により異なる場合がある。この場合、それぞれの機種に固有のプログラムが必要となるため、ソフトウェアの共用・流用が困難である。さらに、多重ループや高頻度に行われる個所に電力制御に関する命令を多数記述した場合は、プログラムの実行速度が低下する。

【0011】

【課題を解決するための手段】本発明の目的は、ハードウェアの規模を増大することなく、またユーザーの開発負担を伴わず、ソフトウェアの汎用性を確保しながら、きめ細かな電力制御を行うことである。

【0012】上記目的は、プログラムのコンパイル、あるいはアセンブル時に静的にプログラムを解析し、この解析結果に基づいて電力制御に関する指示をプログラムに付与する命令変換装置によって達成される。

【0013】命令プログラムを構成する命令の種類により、各動作資源が動作するか否かに関する情報を予め命令別動作資源テーブルに記録しておき、マイクロプロセッサ動作時に所定長の命令区間動作しない動作資源を検出することにより、上記のような命令変換装置が容易に実現される。

【0014】また、電力制御に関する指示を付与する方法としては、電力制御に関する命令を挿入する、あるいは電力制御に関する指示を含まない命令を電力制御に関する指示を含む命令に置換するといった方法が挙げられる。

【0015】命令を挿入することにより低消費電力を行

う方法では命令を多数挿入することによりサイクル数が増えて処理速度が低下することがあり得る。また命令の挿入によりコードサイズが増大する。しかし、これらの課題は上記命令変換装置において、コンパイルあるいはアセンブル時に予めオプションによってユーザーが電力制御のレベルを決定することができる構成にすることにより解決される。すなわち、挿入される電力制御のための命令の数を複数段階で調整することにより、コードサイズと電力制御との調整を行うことができる。

【0016】本発明の別の目的は、実行オブジェクトをデコードする前にこれを解析し、電力制御に関する指示を付与することにより、ハードウェアの増大を一定以下に抑えて低消費電力を達成し得る命令変換回路を提供することである。

【0017】上記目的はマイクロプロセッサが実行する実行オブジェクトを構成する命令コードの種類により、各動作資源が動作するか否かに関する情報を予め記録した命令別動作資源メモリと、前記命令別動作資源メモリの情報に基づき、マイクロプロセッサ動作時に所定命令区間動作しない動作資源を検出する電力制御情報解析回路と、前記命令別動作資源解析回路の検出結果を基に、実行オブジェクトに、電力制御に関する指示を付与する電力制御指示付与回路とを備える命令変換回路によって達成し得る。

【0018】本発明のさらに別の目的は、上記の命令変換装置によって生成された電力最適化済実行オブジェクトを実行するプロセッサにおいて、割り込み・分岐・例外などの異常処理が発生した場合においても、誤動作を生じることなく命令実行可能なプロセッサを提供することにある。

【0019】上記目的は、割り込み・分岐・例外のうち少なくとも1つを検出したときは、電力制御レジスタの状態を全ての動作資源が動作する初期状態へとリセットする状態リセット回路をプロセッサに設けることによって達成される。

【0020】

【発明の実施の形態】今後の説明において、マイクロプロセッサを動作させるための命令列を、その記述言語によらず「命令プログラム」という。また命令プログラムを機械語に翻訳することにより生成されるマイクロプロセッサが実行するデコード前の機械語命令列を「実行オブジェクト」という。なお、「実行オブジェクト」も「命令プログラム」の一種であり、命令プログラムから実行オブジェクトを生成する際にできるあらゆる中間プログラムも命令プログラムに含まれるものとする。

【0021】(第1実施形態) 以下、第1実施形態について、図面を用いて詳細に説明する。

【0022】(命令変換装置の全体構成) 図1は本発明における第1実施形態における命令変換装置を示したものである。図中の101はマイクロプロセッサで実行さ

れる命令プログラムである。ここで101の命令プログラムはニーモニックで記述された命令列あるいはC言語やFORTRAN等の高級プログラム言語で記述された命令列のいずれの表現形式でも構わないが、本実施形態においてはニーモニックで記述された命令列を例にとって説明を行う。

【0023】命令別動作資源テーブル103には、命令毎に、マイクロプロセッサ内のいずれの動作資源が動作するかを情報を予め記録しておく。

【0024】命令別動作資源解析手段102は、命令プログラム101を入力とし、命令プログラム101を構成する命令毎に、命令別動作資源テーブル103を参照して、いずれの動作資源が動作するかを抽出し、命令別動作資源使用データ104を出力する。

【0025】電力制御情報検出手段105は、命令別動作資源使用データ104を入力内容とし、時系列的な解析を行い、いずれの個所にて動作資源の停止あるいは動作開始等の電力制御命令を挿入すると電力消費量を低減できるかを判断し、電力制御指示データ106を出力する。

【0026】電力制御指示付与手段107は、電力制御指示データ106を基に、命令プログラム101に動作資源の停止、あるいは動作開始等の命令の挿入を行い、電力最適化済命令プログラム108を生成する。

【0027】電力制御指示付与手段107によって生成された電力最適化済命令プログラム108の内容は、命令翻訳手段109への入力とし、命令翻訳手段109はマイクロプロセッサが実行すべき機械語命令列である電力最適化済実行オブジェクト110を生成出力する。

【0028】(命令別動作資源テーブルの詳細な構成) 図2を用いて、命令別動作資源テーブル103について説明する。本実施形態では、命令別動作資源テーブル103は命令モードを示すフィールド201とマイクロプロセッサ内の動作資源を表わすフィールド202の2つのフィールドを持つ二次元テーブルである。テーブル中の値は各命令モードの実行において各動作資源が動作するか否かを示している。テーブル中“1”は動作資源が動作すること、“0”は動作資源が動作しないことを示している。例えば、レジスタ間の加算命令である「ADD Rx, Ry, Rz」の場合、マイクロプロセッサ動作時には「演算器A動作」、「データレジスタR0—R15」、「データレジスタR16—R31」のみが動作するのでテーブル中の対応する個所には“1”が記述されており、その他の動作資源は動作しないのでテーブル中の対応する個所には“0”が記述されている。同様に、メモリの読み出し命令である「LD Ra, MEM y」では、テーブル中で“1”が記述されている「メモリRead動作」及び「ブロックA動作」のみが動作することになる。ここで絶対値としての“0”、“1”は重要でなく、動作情報をテーブルとして保有することが

主目的である。

【0029】(命令毎動作資源解析手段の動作説明) 図3を用いて、命令プログラムが301の場合を例にとつて、命令毎動作資源解析手段102の動作説明を行う。図3は本実施形態における命令毎動作資源使用データ104を示している。命令毎動作解析手段102は命令プログラム301を入力として受け取ると、命令毎に命令別動作資源テーブル103の命令モードフィールド201とパターン一致検索を行い、該当命令がどの命令モードに対応しているかを判断して、302のような形で出力する。302において、命令毎に“1”となっている動作資源が、マイクロプロセッサ動作時に動作することを示している。命令毎動作資源解析手段102は、命令プログラムを構成する各命令について上記解析を行い、命令毎動作資源使用データ104を生成する。

【0030】(電力制御情報検出手段の動作説明) 図4を用いて、電力制御情報検出手段105の動作説明を行う。電力制御情報検出手段105は、命令毎動作資源解析手段102が出力する命令毎動作資源使用データ104について、命令方向(時系列方向)に、動作資源が動作するか否かについて解析を行う。解析方法の一例として、連続する“0”を検出する方法を説明する。本実施の形態の命令毎動作資源使用データ104において、情報“0”はマイクロプロセッサがその命令実行において動作資源の動作を行わないことを示す。電力制御情報検出手段105は、命令毎動作資源使用データ104について、ある一定命令区間、例えば3命令区間以上連続する“0”を検出することにより、所定命令区間動作しない動作資源を検出できる。

【0031】「メモリRead動作」の解析を例にとつて、3命令区間以上動作しない動作資源を検出する場合について説明する。「メモリRead動作」に対応する401に示したデータ列によると、SUB命令からSTR命令まで4命令の間、「メモリRead動作」を必要としない。従って電力制御情報検出手段105はこれを検出する。検出された動作資源と停止可能区間に関する情報は電力制御指示データ106として電力制御指示付与手段107へと出力される。

【0032】(電力制御指示付与手段の一例) 図4と同じ命令プログラムを例にとつて、電力制御指示付与手段107について説明を加える。図5は電力制御命令の挿入例である。図5に示すように、電力制御指示付与手段107は、電力制御情報検出手段105が出力する電力制御指示データ106に基づき、SUB命令の直前に「メモリRead動作」に関する回路停止を要求する命令コードを、STR命令の直後に回路動作開始の命令を挿入している、これにより、この区間において命令毎のデコードを必要とせず、「メモリRead動作」に関する機能停止を行うことができ、「メモリRead動作」に必要とされた消費電力を削減することができる。

【0033】なお図4の例では、SUB命令からMUL命令までの3命令間は「メモリWrite動作」も必要としない。従って本実施の形態では、この情報を検出することで、動作資源個別の電力制御のみを行う場合よりも、さらに上流側で電力制御を行うことができる。すなわち、命令プログラムに「メモリRead動作」や「メモリWrite動作」に関する制御指示を付与するのみならず、メモリやメモリ制御回路自体を完全に停止させる制御指示を付与することもできる。

【0034】上記の命令変換装置はコンピュータプログラムとして実現され、アセンブラプログラムとしてマイクロプロセッサのユーザーに提供される。ユーザーは命令プログラム101をコンパイルあるいはアセンブルすれば、電力最適化済実行オブジェクト110までを作成するようにできる。これにより、ユーザーは何も意識する必要なく、ただ命令プログラムを記述し、アセンブルするだけでマイクロプロセッサの消費電力は従来のものより低減される。

【0035】(マイクロプロセッサにおける電力制御方法) 上記の命令変換装置によって作成された電力最適化済実行オブジェクト110を実行することにより従来と比べて電力消費量は低減される。図6は本実施形態におけるマイクロプロセッサの構成を示した図である。

【0036】命令記憶装置602には本実施形態における命令変換装置で作成された電力最適化済実行オブジェクト110が記憶されている。プログラムカウンタ601はマイクロプロセッサにおいて次に実行される命令のアドレスを指定する。命令デコーダ603は、プログラムカウンタ601で指定されたアドレスの命令コードを命令記憶装置602から取得し、デコードする。命令デコーダ603でデコードされた電力制御に関する指示は、電力制御レジスタ604に書き込まれ(608)、この電力制御レジスタ604の状態により電力制御実行装置605がマイクロプロセッサ内の電力制御(クロック停止、クロック周波数変更、電源遮断、電圧変更、信号変化抑制等)を行う。例えばデータメモリを停止する指示を命令デコーダ603がデコードしたならば、データメモリを停止する指示を電力制御レジスタ604に書き込む。するとマイクロプロセッサは、命令実行前に電力制御レジスタ604の状態を参照し、データメモリを停止することができる。停止後に再びメモリを使用するならば、電力制御レジスタ604に再度、停止解除の指示を書き込む。するとマイクロプロセッサは命令実行前に電力制御レジスタ604の状態を参照し、メモリ機能の停止を解除する。

【0037】すなわち、電力最適化済実行オブジェクト110を命令デコーダ603に送り、電力制御に関する指示を取り出して、電力制御レジスタ604に書き込むことにより、ハードウェアの規模を増大させることなく、きめ細かな電力制御が行え、マイクロプロセッサの

消費電力を低減させることができる。

【0038】ここでマイクロプロセッサに特殊な命令が与えられた場合には、電力最適化済実行オブジェクト110を実行することにより、マイクロプロセッサが誤動作を引き起こすおそれがある。以下にその詳細について説明する。

【0039】通常のプログラム実行時には、予め命令の実行シーケンスが確定しているため、電力最適化済実行オブジェクト110をデコードすることにより、電力制御レジスタ604に正確に電力制御に関する指示を書き込むことができる。しかし、割り込み・分岐・例外のような特殊な命令が発生した場合には、プログラムの命令実行シーケンスが通常と異なり、プログラムが記述された順番に実行されるわけではない。すなわち、プログラムの記述された順番に従うなら次はADD命令が投入されるような場合であっても、割り込みによってSTR命令が投入されるようなことがあり得る。この場合、電力最適化済実行オブジェクト110をデコードすることにより、電力制御レジスタ604がメモリ停止状態に設定されていても、割り込み・分岐・例外によりメモリを使用する命令が投入される可能性がある。そうすると、メモリは動作することができずにマイクロプロセッサは正常動作を行わないおそれがある。

【0040】これを防止するために、本実施の形態のマイクロプロセッサにおいては状態リセット回路606を設けている。これによって、特殊な命令が与えられた場合でも、誤動作を引き起こすことなく、きめ細かな電力制御を行うことが出来る。

【0041】状態リセット回路606は割り込み・分岐・例外の要因信号607を検出すると、電力制御レジスタ604の状態を、全ての動作資源が動作する状態（以降、既定値という）に設定する。要因信号607としては、演算結果の例外処理信号（例えば、演算結果のオーバーフロー時に発生する例外要求信号）、プロセッサ自身の割り込み要求信号（例えば、DMA転送割り込みやタイマー割り込み等、内容に応じて要求する信号）、外部からの入力信号（例えば、外部端子の変化を要求信号とする）等が挙げられる。電力制御レジスタ604が既定値に設定されることにより、マイクロプロセッサは全ての動作資源を動作させることができ、マイクロプロセッサは割り込み・分岐・例外があった場合にも誤動作を起こさない。

【0042】以上、本発明の第1実施形態について説明したが、本発明はこの実施形態に限られないことはもちろんである。

【0043】第1実施形態に係る命令変換装置は、ユーザーの記述言語がC言語やFORTRANのような高級プログラム言語である場合にはコンパイラとしてユーザーに提供される。高級プログラム言語を用いることによりユーザーはより簡単にプログラムを記述することが

きる。

【0044】また、第1実施形態に係る命令変換装置は、アセンブラやコンパイラとは別の電力制御プログラムとして提供し、ユーザーがこの電力制御プログラムを実行することにより、まず電力最適化済命令プログラム108を作成し、その後アSEMBルすることにより電力最適化済実行オブジェクト110を作成するようにすることも可能である。この場合、ユーザーが記述した命令プログラム101と、電力最適化済命令プログラム108との比較を行うことにより、どの個所に電力制御命令が付与されているかをユーザーは容易に確認することができる。

【0045】電力制御情報解析手段111は上記の実施形態に限られるものではなく、命令プログラムを静的に解析することにより電力制御可能な命令区間を検出できるものであれば何でも良い。例えば、電力制御情報検出手段105で検出される電力制御を必要とする命令区間は3区間に限られるのではなく、1区間以上の長さで、マイクロプロセッサで動作する各アプリケーションプログラムやシステムに最適な電力制御のレベルによって任意に設定することができる。

【0046】電力制御指示付与手段107は上記のように命令挿入を行うものには限られず、電力制御指示データ106に基づいて電力制御情報を付与するものであれば何でも構わない。例えば電力制御指示付与手段107は図7のような形態でも良い。すなわち、図5では、電力制御指示付与手段107において新たな命令を挿入する実施形態を示したが、予め電力制御情報を含んだ命令をプロセッサの命令仕様において定義しておけば、図7に示すように、既存の一命令を電力制御情報を含んだ別種の命令に置換することができる。この場合、マイクロプロセッサが動作する際に、命令追加挿入による新たな動作サイクルの増加は発生しないため、速度低下を招かずに電力制御を実現できる。

【0047】もちろん、電力制御情報を含んだ命令が定義されているならば置換を行い、定義されていない命令に対しては挿入を行うようにしても良い。

【0048】状態リセット回路606はマイクロプロセッサが正常動作を行うようにするものであれば何でも構わない。例えば、第1実施形態では割り込み・分岐・例外を検出しているが、これは割り込み・例外のみを検出するものでも良い。この場合であっても、分岐命令直後は電力制御レジスタをリセットする命令を挿入するように、命令変換装置を設定しておけばマイクロプロセッサは正常動作を行うことができる。

【0049】また、状態リセット回路606は、命令デコードのデコード結果609から、割り込み・分岐・例外を検出するものであっても良い。

【0050】また、状態リセット回路606は、命令デコードのデコード結果609と、要因信号607を併用

して、割り込み・分岐・例外を検出しても良い。両方を併用することにより、どちらか一方のみを用いるよりも確実に割り込み・分岐・例外の検出を行うことができる。

【0051】また、図8に示すように、状態リセット回路606はプログラムカウンタの状態変化から割り込み・分岐・例外の検出を行うものであっても良い。

【0052】(第2実施形態)本発明における第2実施形態について、図面を参照しながら説明する。

【0053】図9は第2実施形態に係る命令変換装置を示した図である。図9において図1と共通の構成要素には同一の符号を付している。電力制御のための命令を挿入することにより、マイクロプロセッサに最終的に実装される機械語プログラムのコードサイズは増大する。また、命令の実行サイクルも増加しプロセッサの処理速度の妨げとなる。ユーザーによっては低消費電力よりもコードサイズや処理速度を優先したい場合もありうる。そこで第2実施形態においては、ユーザーがアセンブルあるいはコンパイルを行うときに、オプションによって電力制御のレベルを選択できるような構成としている。ユーザーはアセンブルあるいはコンパイルを行うときにオプション“—P1”又は“—P2”をつけてアセンブルあるいはコンパイルを行う。ユーザーが入力したオプションはオプションレジスタ902に格納される。電力制御レベル判定手段901はオプションレジスタ902の値を参照し、参照結果を電力制御情報検出手段105へと出力する。電力制御情報検出手段105はオプションが“—P1”のときは3区間以上動作を行わない動作資源を検出し、オプションが“—P2”のときは5区間以上動作を行わない動作資源を検出して電力制御指示データ106を作成する。

【0054】よって、例えば図4のような命令毎動作資源使用データ104を例にとって説明すると、オプションが“—P1”のときは“メモリRead動作”は3区間以上動作しない(SUB命令からSTR命令までの間)動作資源として検出されるが、オプションが“—P2”のときは、“メモリRead動作”は検出されない。

【0055】オプションによって電力制御のレベルをユーザーが任意に選択できる構成にすることにより、挿入される電力制御のための命令の数を調整できるので、ユーザーはコードサイズや実行サイクルを考慮に入れながら最適な電力制御のレベルを設定することができる。

【0056】電力制御レベル判定手段901は上記のようにオプションによって電力制御レベルを判定する構成に限られるものではなく、ユーザーの判断で電力制御レベルが複数段階の中から設定できるものであれば何でも構わない。また、第2実施形態において電力制御情報検出手段105は停止可能と判断する命令区間の長さを変えることにより電力制御レベルを調整しているが、電力

制御レベルを調整する手法はこれに限られるものではない。例えば、オプションが“—P1”のときは全ての動作資源に対して検出を行い、オプションが“—P2”のときはメモリとデータレジスタのみについて検出を行うようにしても良い。すなわち、電力制御情報検出手段105は電力制御レベルに応じて、電力制御の対象とする動作資源を変化させるものであっても良い。

【0057】また、電力制御指示付与手段107が命令挿入と置換を併用可能な場合には、オプションが“—P1”のときは挿入も置換も行い、オプションが“—P2”のときは置換のみを行うようにしても良い。置換によってはコードサイズ、実行サイクルの増大は生じないのでオプション“—P2”を用いることによって、デメリットを伴わない最小限の電力制御のみが行われる。

【0058】(第3実施形態)本発明における第3実施形態について、図面を参照しながら説明する。

【0059】図10は本発明における第3実施形態の命令変換装置を示したものである。図10において、図1と共通の構成要素には図1と同一の符号を付している。第1実施形態ではユーザーが記述した命令プログラムに、電力制御に関する命令を挿入あるいは置換することにより電力最適化済命令プログラム108を作成していたが、本実施の形態においてはユーザーが記述した命令プログラム101は、まず命令翻訳手段109において、マイクロプロセッサが実行する機械語命令列である実行オブジェクト1001に翻訳される。その後、実行オブジェクト1001を入力として、電力最適化済実行オブジェクト110が生成される。

【0060】この場合、図11に示すように、電力制御指示付与手段(図10の107)では機械語で記述された命令列の一部分に電力制御に関するビットを挿入することにより、電力制御に関する指示を命令コードに付加することができる。この場合においてユーザーはそのまま電力制御指示がどの個所に付与されたかを知るのは困難であるが、電力最適化済実行オブジェクト110をデアセンブルすることにより電力制御指示がどの個所に付与されたかを知ることができる。

【0061】第3実施形態において電力制御指示付与手段107は上記のビット挿入を行うものには限られず、電力制御指示データ106に基づいて、電力制御情報を付与できるものであれば何でも構わない。例えば、機械語命令自体を挿入するものであっても良い。

【0062】第3実施形態における命令変換装置も第1実施形態と同様にコンピュータプログラムにより実現することができる。アセンブラ・コンパイラとしてマイクロプロセッサのユーザーに提供しても良いし、単独の電力制御プログラムとしても良い。

【0063】また、第2実施形態のように電力制御レベル判定手段を備え、電力制御レベルをユーザーが任意に選択できるようにしても良い。

【0064】また、第3実施形態の命令変換装置は実行オブジェクトをそのまま変換するものであるから、ユーザーのプログラムの記述言語には依存せず、より汎用性の高い、電力最適化を行うコンパイラ・アセンブラとしてユーザーに提供できる。

【0065】また、第3実施形態におけるマイクロプロセッサの構成は第1実施形態と同様に状態リセット回路606を備えた構成を用いることができるというまでもない。

【0066】(第4実施形態) 本発明における第4実施形態について、図面を参照しながら説明する。

【0067】図12は本発明における第4実施形態の命令変換回路の構成図である。ここで命令コード供給装置1200は、従来の方法で生成された電力最適化されていない実行オブジェクトをマイクロプロセッサの命令記憶装置1205に送る装置である。命令記憶装置1205はフラッシュ等のように書き換え可能なものでも、書き換え不可能なROMでもかまわない。ここで従来の方法であれば、実行オブジェクトは、直接命令記憶装置1205に送られていたが、本実施の形態においては、本発明の命令変換回路1207を通り、電力最適化済実行オブジェクトに書き換えられた後、命令記憶装置1205に送られる。命令変換回路1207は命令別動作資源メモリ1201、命令毎動作資源解析回路1202、電力制御情報検出回路1203及び電力制御指示付与回路1204から構成されている。命令別動作資源メモリ1201は第1実施形態における命令別動作資源テーブル103を記憶したメモリである。命令毎動作資源解析回路1202とは第1実施形態における命令毎動作資源解析手段102に対応するもので、命令別動作資源メモリ1201に記憶されている命令別動作資源テーブルを参照して、命令コード毎に動作資源の動作・停止を判断し、命令毎動作資源使用データを出力する。電力制御情報検出回路1203は第1実施形態の電力制御情報検出手段105に対応するもので、命令毎動作資源解析回路1202が出力する命令毎動作資源使用データに基づいて、所定命令区間の間停止している動作資源を抽出し、電力制御指示データを出力する。電力制御指示付与回路1204は第1実施形態の電力制御指示付与手段107に対応するもので、電力制御情報検出回路1203から出力される電力制御指示データに基づいて、命令コード供給装置1200から命令記憶装置1205に供給される実行オブジェクトに電力制御に関する指示を付与し、電力最適化済実行オブジェクトを生成する。命令記憶装置1205には電力最適化済実行オブジェクトが保存される。この電力最適化済実行オブジェクトを実行することにより、マイクロプロセッサは消費電力を低減することができる。

【0068】上記のように本発明の命令変換方法は回路によっても実現できる。

【0069】(第5実施形態) 本発明における第5実施形態について、図面を参照しながら説明する。

【0070】図13は本発明における第5実施形態の命令変換回路の構成図である。図13において、図12と共通の構成要素には図12と同一の符号を付している。第4実施形態では命令コード供給装置1200から命令記憶装置1205に実行オブジェクトが保存される際に電力制御指示の付与を行っていたが、本実施形態では命令記憶装置1205に保存された実行オブジェクトは、本発明の命令変換回路1207に直接送られ、電力制御に関する指示を付与されたうえで、再び命令記憶装置1205に保存される。第3実施形態と異なり、命令記憶装置1205にすでに保存されている実行オブジェクトに対しても電力制御に関する指示を付与することができる。また、本実施形態において命令記憶装置1205は書き換え可能なものでなければならない。

【0071】なお、第4実施形態から第5実施形態までにおいても、状態リセット回路を設けることにより、特殊な命令シーケンスが発生してもマイクロプロセッサが正常動作を行うようにすることができる。

【0072】

【発明の効果】 本発明によれば、本来処理すべきアプリケーションプログラムの開発と、電力制御を独立に行うことができ、ユーザーのソフトウェア開発の負担は低減される。また、ハードウェアの増大を招くことなく、きめ細かな電力制御をソフトウェアの汎用性を確保したまま容易に実現することができる。

【図面の簡単な説明】

【図1】 本発明における第1実施形態の命令変換装置を示す図

【図2】 本発明における命令別動作資源テーブルの実施の形態を示す図

【図3】 本発明における命令毎動作資源使用データの実施の形態を示す図

【図4】 本発明における電力制御情報検出手段の実施の形態を示す図

【図5】 命令プログラムに電力制御命令を挿入する例を示す図

【図6】 本発明における第1実施形態のマイクロプロセッサの構成を示す図

【図7】 命令プログラム中の一の命令を電力制御指示を含む一の命令に置換する例を示す図

【図8】 本発明における第1実施形態のマイクロプロセッサの別の形態を示す図

【図9】 本発明における第2実施形態の命令変換装置を示す図

【図10】 本発明における第3実施形態の命令変換装置を示す図

【図11】 本発明における第3実施形態で機械語命令への電力制御情報の付与を示す図

【図12】本発明における第4実施形態の命令変換回路を示す図

【図13】本発明における第5実施形態の命令変換回路を示す図

【図14】従来のハードウェア駆動における電力制御方法を示す図

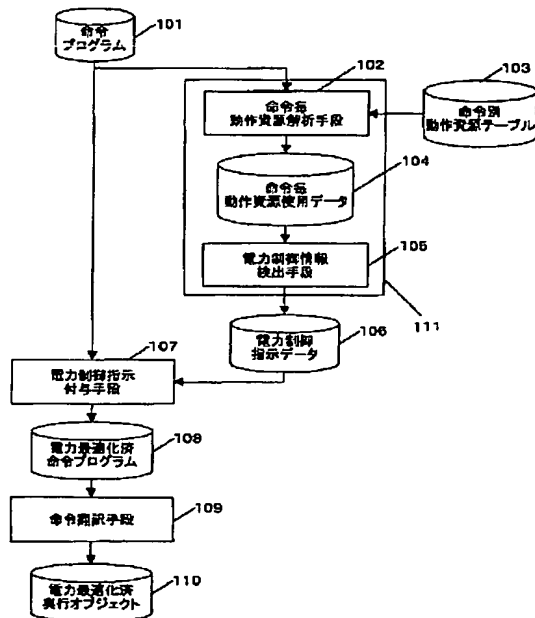
【図15】従来のソフトウェア駆動における電力制御方法を示す図

【符号の説明】

- 101 命令プログラム
- 102 命令毎動作資源解析手段
- 103 命令別動作資源テーブル
- 104 命令毎動作資源使用データ
- 105 電力制御情報検出手段
- 106 電力制御指示データ
- 107 電力制御指示付与手段
- 108 電力最適化済命令プログラム
- 109 命令翻訳手段
- 110 電力最適化済実行オブジェクト
- 111 電力制御情報解析手段

- 201 命令モードフィールド
- 202 動作資源フィールド
- 601 プログラムカウンタ
- 602 命令記憶装置
- 603 命令デコーダ
- 604 電力制御レジスタ (PCR)
- 605 電力制御実行装置
- 606 状態リセット回路
- 607 要因信号
- 901 電力制御レベル判定手段
- 902 オプションレジスタ
- 1200 命令コード供給装置
- 1201 命令別動作資源メモリ
- 1202 命令毎動作資源解析回路
- 1203 電力制御情報検出回路
- 1204 電力制御指示付与回路
- 1205 命令記憶装置
- 1206 命令デコーダ
- 1207 命令変換回路
- 1208 電力制御情報解析回路

【図1】



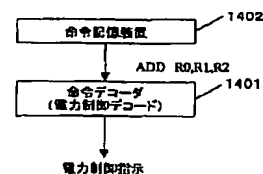
【図2】

201 命令モード		202 動作資源															
		メモリ Read 動作	メモリ Write 動作	演算器 A 動作	演算器 B 動作	分岐ユニット A 動作	ブロック A 動作	ブロック B 動作	ブロック C 動作	周辺機能/バス動作	並列命令デコードユニット	デマルチプレクサ R0-R15	デマルチプレクサ R16-R31	アドレス変換ユニット	即値ユニット		
ADD R _x ,R _y ,R _z		0	0	1	0	0	0	0	0	0	0	0	1	1	0	0	
ADD R _x ,MEM _y ,MEM _z		1	0	1	0	0	1	0	0	0	0	0	1	1	1	0	
ADD R _x ,R _y ,imm		0	0	1	0	0	0	0	0	0	0	1	0	0	0	1	
LD R _x ,MEM _y		1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	
LD R _x ,MEM _y :ADD R _x ,R _y ,R _z		1	0	1	0	0	1	0	0	0	1	1	1	0	0	0	
STR R _x ,MEM _y		0	1	0	0	0	1	1	0	0	0	1	0	0	0	0	
MUL R _x ,R _y ,R _z		0	0	0	1	0	0	0	0	0	0	0	1	1	0	0	
IO ADDR		0	0	0	0	0	0	0	0	1	0	0	0	1	1		
JLMP R _x		0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	
LOOP N		0	0	0	0	1	0	0	1	0	0	0	0	0	0	1	
		:															

命令別動作資源テーブル

命令別動作資源テーブル

【図14】



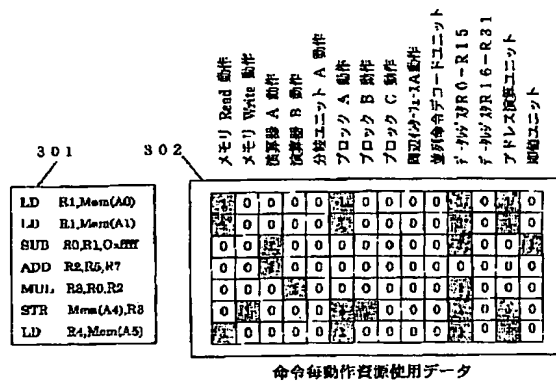
【図5】

```
LD R1,Mem(A0)
LD R1,Mem(A1)
SUB R0,R1,0x00FF
ADD R2,R5,R7
MUL R3,R0,R2
STR Mem(A4),R3
LD R4,Mem(A5)
```

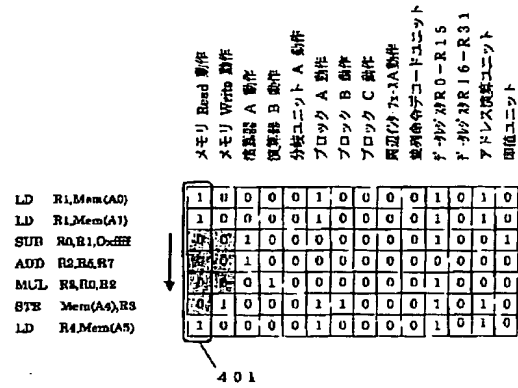
【図7】

```
LD R1,Mem(A0)
LD R1,Mem(A1)
SUB R0,R1,0x00FF
ADD R2,R5,R7
MUL R3,R0,R2
STR Mem(A4),R3
LD R4,Mem(A5)
```

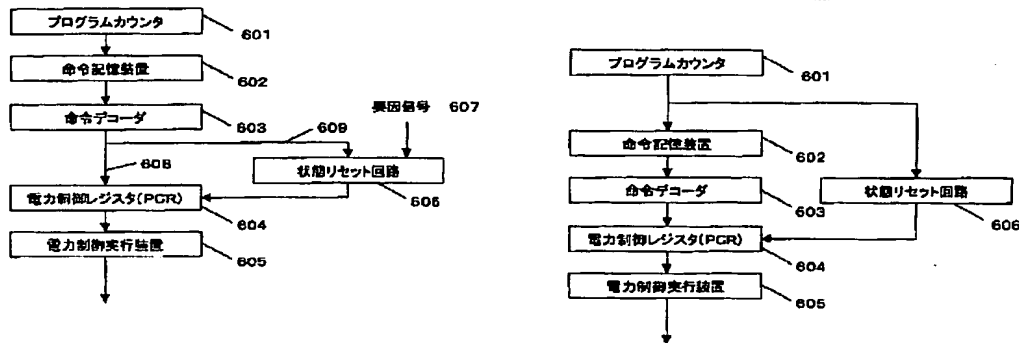
【図3】



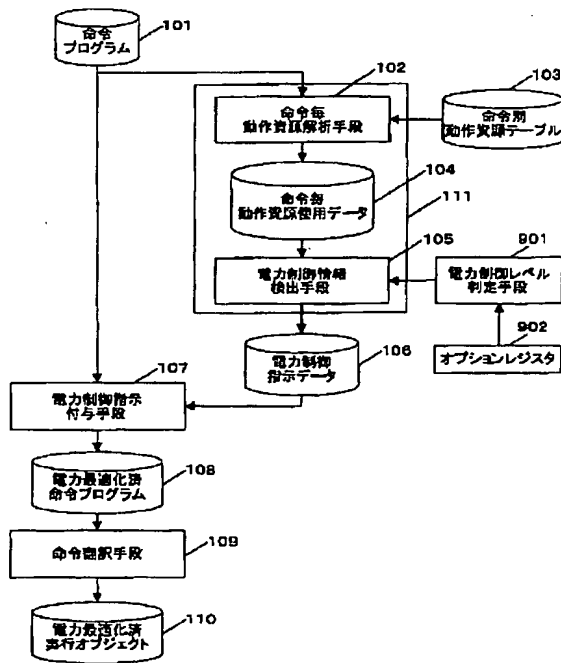
【図4】



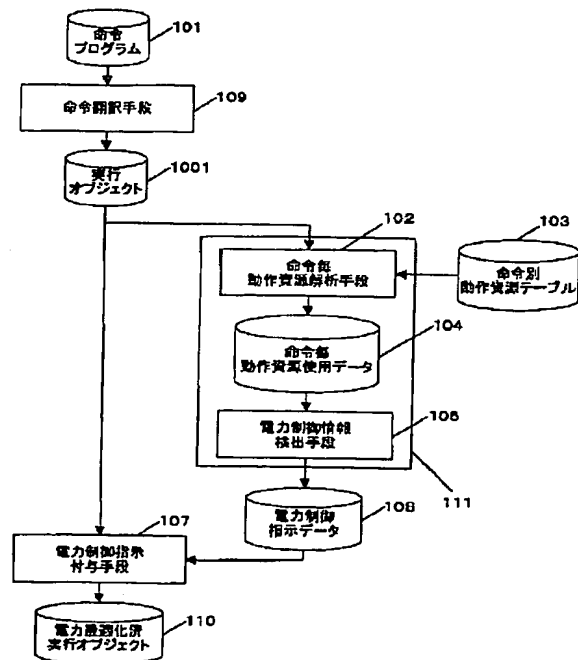
【図6】



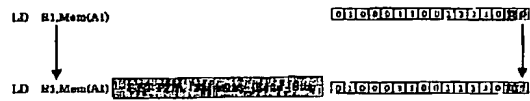
【図9】



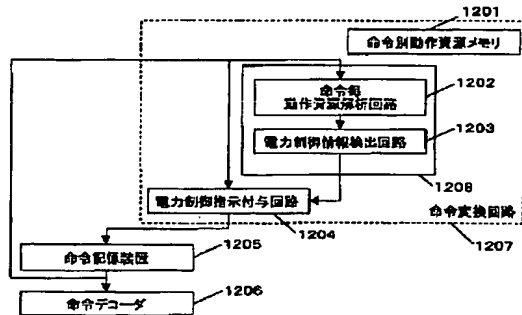
【図 10】



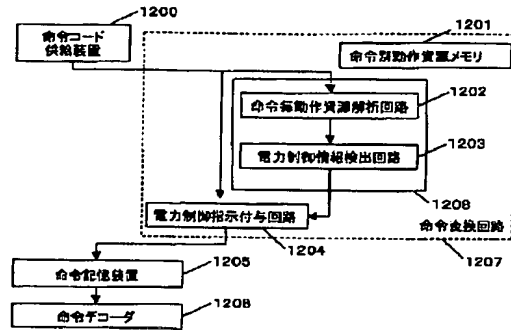
【図11】



【図13】



【図12】



【図15】

